Claims:

What is claimed is:

1. A system for providing access to web services, comprising:

a container driver that accepts invoke requests from a client for web services,

an interceptor that receives context information for the invoke request from said container driver, and modifies the message context to be used with web services; and,

an invocation handler that receives the modified context information from said container driver, passes parameters from the message context to the target of the request, processes values returned from the target, and passes the values to the container driver, such that the container driver can formulate a response to the invoke request.

2. The system of claim 1 wherein the client utilizes JAX-RPC to invoke the web services.

3. The system of claim 1 wherein said container driver is adapted to perform any data binding and unbinding required to process the invoke request.

4. The system of claim 1, further comprising a protocol adapter that intercepts web service invoke requests and passes the web service invoke requests to said container driver.

5. The system of claim 4, wherein said protocol adapter converts the format of an invoke request and create a message context containing the invoke request.

6. The system of claim 1, further comprising a plugin component to be used by said container driver to perform any data binding and unbinding.

7. The system of claim 1, further comprising an invocation context for storing arbitrary context data useful in processing the web request, said invocation context available to at least one of said interceptor and said invocation handler.

8. The system of claim 1, wherein said invocation handler manages security policies, transaction management, and target object life cycle for the request.

9. The system of claim 1, further comprising a web service container for hosting said container driver, said interceptor, and said invocation handler.

10. The system of claim 1, further comprising a target object to which said invocation handler can delegate processing the invoke request.

11. A method for providing access to web services, comprising:

receiving at a container manager an invoke request from a client to access web services;

formatting message context for the invoke request to be used with web services;

binding the message context;

processing the request using an invocation handler and generating response data;

unbinding the message context containing the response data; and,

reformatting the message context for responding to the invoke request.

12. The method of claim 11 wherein the client utilizes JAX-RPC to invoke the web services.

13. The method of claim 11 wherein a container driver is used to perform any data binding and unbinding required to process the invoke request.

14. The method of claim 11, further comprising intercepting an invoke request from a web services client using a protocol adapter and generating message context for the invoke request to be sent to the container manager.

15. The method of claim 11, wherein said step of formatting message context comprises using an interceptor to format the message context.

16. The method of claim 11, wherein said step of binding the message context comprises using a codec selected from the group consisting of Java Binding codecs, SOAP codecs, XML codecs, and custom codecs.

17. The method of claim 11, further comprising storing arbitrary context data for use in processing the invoke request.

18. The method of claim 11, further comprising managing life cycle, transaction, and security information for the processing of the invoke request.

19. The method of claim 11, further comprising delegating the processing of the invoke request to a target object.

20. A computer readable medium, including instructions stored thereon which when executed by the computer cause the computer to perform the steps of:

receiving at a container manager an invoke request from a client to access web services;

formatting message context for the invoke request to be used with web services;

binding the message context;

processing the request using an invocation handler and generating response data;

unbinding the message context containing the response data; and,

reformatting the message context for responding to the invoke request.